Super Stack

Given a stack, empty at the start, and a series of operations on that stack, you need to take a "peek" at the value on top of the stack and print it to output after every operation. If stack is empty then print "EMPTY" (without quotes).

Allowable Operations:

`push a` // Push an element with value *a* to the top of the stack
`pop` // Pop the top element from the stack
`inc x d` // Add the value ***d*** to each of the bottom ***x*** elements on the stack

Input Format:

n - Integer, number of operations, on the top line.
n lines with exactly one operation per line from the list above.

Output Format:

n lines of text, with the element at the top of the stack on each line.
If stack is empty, then print "***EMPTY***" *(without quotes).*
*Note: For a **pop** operation, the output is the element exposed by the **pop,** not the element popped off.*

Constraints:

-1,000,000,000 ≤ ***a*** ≤ 1,000,000,000
-1,000,000,000 ≤ ***d*** ≤ 1,000,000,000
0 ≤ ***n*** ≤ 500,000
1 ≤ ***x*** ≤ size of the stack at the time of the operation
Assume there is no "pop" operation when the stack is empty.

Note: Efficiency is going to be crucial to pass some of the largest test cases. (If you are using Java, think about improving the IO as well, e.g: using PrintWriter)

Sample Input:

```
12
push 4
pop
push 3
push 5
push 2
inc 3 1
pop
push 1
inc 2 2
push 4
pop
```



```
4
EMPTY
3
5
2
3
6
1
1
4
1
8
```

Explanation

(The state of the whole stack is shown in square brackets, bottom to top.)

push 4 [4]
pop [] - Since list is empty, print "EMPTY" after this operation
push 3 [3]
push 5 [3, 5]
push 2 [3, 5, 2]
inc 3 1 [4, 6, 3]
pop [4, 6]
push 1 [4, 6, 1]
inc 2 2 [6, 8, 1]

push 4 [6, 8, 1, 4]
pop [6, 8, 1]
pop [6, 8]

YOUR ANSWER

Java 7

Click here to know how to read from STDIN and write to STDOUT

```
1  import java.io.*;
2  public class Solution {
3      public static void main(String args[] ) throws
   Exception {
4          /* Enter your code here. Read input from STDIN.
   Print output to STDOUT */
5      }
6  }
```

Line: 1 Col: 1

Test against custom input

Run Code

Submit code & Continue

Download sample testcases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*